

혼합 분포 확률 시간 넷을 이용한 웹 서비스 시스템의 응답 시간 분석

임재걸[†], 심규박^{**}, 도재수^{***}

요 약

오늘날, 대부분의 고객들이 어디에서든지 인터넷을 사용할 수 있게 됨에 따라, 대부분의 기업들이 자신들의 서비스를 인터넷상에서 제공하게 되었다. 그 결과 무수히 많은 웹 서비스 시스템이 이미 인터넷상에서 서비스를 하고 있으며, 더 많은 시스템들이 구축 중이다. 그래서 구축 중인 시스템이 교착상태와 같은 문제점이 없이 성공적으로 실행될 것이라는 것을 설계 초기 단계에 검증하는 방법에 대한 연구가 많이 수행되어 왔다. 페트리 넷을 이용하는 방법들도 몇 가지 소개되었는데, 이들은 웹 서비스 시스템을 페트리 넷으로 표현하는 방법에 주안점을 두고 있으며, 효율적인 분석 방법에 대한 연구는 부족한 실정이다. '최소 순회 시간' 방법은 '시간 넷'(Timed Net)에서 모든 트랜지션(transition)들을 최소한 한번 이상 격발하고 초기 마킹으로 되돌아오는데 걸리는 최소 시간을 찾아주는 수학적 방법으로, 컴퓨터 시스템 분석에 널리 사용된다. 시간 넷은 트랜지션에 지연 시간이 연관된 변형된 페트리 넷으로, 실세계에서의 지연시간은 확률적임에도 불구하고 기존의 시간 넷에서는 상수가 사용된다. 본 논문은 사건의 실행 시간이 확률 분포를 이루는 '혼합 분포 확률 시간 넷'을 제안하고 '혼합 분포 확률 시간 넷'의 최소 순회 시간 분석 방법을 소개한다. 또한 '혼합 분포 확률 시간 넷'의 최소순회시간 분석을 이용하여 웹 서비스 시스템의 응답 시간을 분석하는 방법을 보인다.

Response Time Analysis of Web Service Systems with Mixedly Distributed Stochastic Timed Net

Jae-Geol Yim[†], Kyu-Bark Shim^{**}, Jae-su Do^{***}

ABSTRACT

Today, consumers can access Internet from everywhere, therefore most commercial and other organizations provide their services on the Web. As the result, countless Web service systems are already on the Internet and more systems are under construction. Therefore, many researches of verifying that the system to be constructed will not have any deadlock and will run successfully without any problem at the early stage of design have been performed. Several Petri net based verification methods have also been published. However, they have focused on building Petri net models of Web service systems and none of them introduces efficient analysis methods. As a mathematical technique with which we can find the minimum duration time needed to fire all the transitions at least once and coming back to the initial marking in a timed net, the minimum cycle time method has been widely used in computer system analysis. A timed net is a modified version of a Petri net where a transition is associated with a delay time. A delay time used in a timed net is a constant even though the duration time associated with an event in the real world is a stochastic number in general. Therefore, this paper proposes 'Mixedly Distributed Stochastic Timed Net' where a transition can be associated with a stochastic number and introduce a minimum cycle time analysis method for 'Mixedly Distributed Stochastic Timed Net'. We also introduce a method of analysing a Web service system's response time with the minimum cycle time analysis method for 'Mixedly Distributed Stochastic Timed Net'.

Key words: Petri net(페트리넷), Timed net(시간넷), System Analysis(시스템분석), Web Service System(웹서비스시스템), Response Time(응답시간)

※ 교신저자(Corresponding Author) : 심규박, 주소 : 경북 경주시 석장동 707(780-714), 전화 : 054)770-2245, FAX : 054)770-2520, E-mail : gpshim@dongguk.ac.kr

접수일 : 2006년 8월 7일, 완료일 : 2006년 9월 20일

[†] 정회원, 동국대학교 과학기술대학 컴퓨터멀티미디어학과 교수 (E-mail : yim@dongguk.ac.kr)

^{**} 정회원, 동국대학교 과학기술대학 정보통계학과 교수

^{***} 정회원, 동국대학교 과학기술대학 컴퓨터멀티미디어학과 교수 (E-mail: dojesu@dongguk.ac.kr)

※ 이 논문은 2005년도 정부(교육인적자원부)의 지원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-041-D00633)

1. 서 론

컴퓨터가 어느 곳이나 산재해 있는 유비쿼터스 시대의 길목에 있는 오늘날, 탁상용 컴퓨터뿐만 아니라 PDA와 휴대용 전화기를 통하여 언제 어디서든 인터넷을 접속하여 방대한 정보를 이용하고 필요한 업무를 처리하는 것이 현대인의 전형적인 모습으로 되어가고 있다. 이에 따라 대부분의 기업들이 자신의 서비스를 앞 다투어 인터넷상에 올려놓는 실정이다. 인터넷에서 URL(Universal Resource Locator)로 접속하여 사용할 수 있는 자동화된 응용소프트웨어를 웹 서비스라고 일컬으며, 교통편예약, 고객 서비스, 회계, 호텔예약, 구매시스템 등 헤아릴 수 없이 많은 유형의 웹 서비스가 이용되고 있다.

웹 서비스가 자동화된 응용소프트웨어라 함은 이들이 비록 서로 다른 플랫폼 상에 서로 다른 언어로 구현되었다 하더라도 표준화된 방법으로 서로 통신이 가능하고 필요에 따라 서로 호출하여 사용할 수 있다는 것이다. 예를 들어 자동차 판매 웹 서비스는 은행이 제공하는 웹 서비스를 호출하여 자동차 구입 고객에게 대출을 제공하며, 은행의 웹 서비스는 신용조회 웹 서비스를 호출하여 고객에게 알맞은 이자율을 산출해서 대출에 적용한다.

이와 같이 다른 웹 서비스를 호출하여 사용하는 웹 서비스 시스템 구축을 가능하게 하는 도구로 SOAP(Simple Object Access Protocol), WSDL(Web Service Description Language), UDDI(Universal Description, Discovery, and Integration) 등이 사용된다. SOAP는 웹 서비스 간에 HTTP를 이용하여 XML 메시지를 주고받는 표준이며 [1], WSDL은 웹 서비스가 제공하는 기능, 있는 곳, 호출 방법 등을 묘사하기 위한 XML 언어이다 [2]. 한편, UDDI는 세계적인 웹 서비스 등록소와 웹 서비스 디렉터리에 웹 서비스 정보를 출판할 때 따라야 하는 표준이다[3].

위와 같은 방법을 이용하여 구축하는 여러 웹 서비스들을 이용하는 웹 서비스 기반 응용소프트웨어 시스템은 사용자에게 매우 환상적인 서비스를 제공할 수 있지만, 동시에 시스템이 너무 비대해짐에 따라 반응 시간이 너무 길어져 사용자로 하여금 불편을 느끼게 하는 결과를 낼 수도 있고, 나아가서 웹 서비스들이 서로 서비스를 요구하고 결과를 기다리는 교착상태에 빠지는 경우도 발생할 수 있다.

따라서 웹 서비스 시스템을 구축하기에 앞서, 구

축하는 시스템이 과연 교착상태와 같은 문제점이 없이 잘 실행될 것인지를 검토하는 방법에 대한 연구가 여러 곳에서 수행되었다. 특히 페트리 넷을 이용하는 방법들[4-9]이 소개되는데, 페트리 넷이 활용되는 초기에는 항상 그렇듯이, 이들은 웹 서비스 시스템을 페트리 넷으로 표현하는 방법에 주안점을 두고 효율적인 분석 방법에 대한 연구는 부족한 실정이다. 즉, 이들이 사용하는 분석 방법은 모든 경우를 다 고려하여보는 기초적인 방법을 적용한다.

본 논문은 혼합 확률분포 시간 페트리 넷을 제안하고, 이것의 최소 순회 시간 분석 방법을 소개한다. 제안하는 방법의 효용성을 보이기 위하여 웹 서비스 시스템을 제안하는 방법으로 분석하는 예를 보인다. 본 논문이 제안하는 혼합 확률분포 시간 페트리 넷에는 처리 과정에 소요되는 시간을 확률적으로 표현할 수 있으므로 웹 서비스 시스템을 더욱 현실적으로 모델링 할 수 있다는 장점이 있다.

2. 관련연구

2.1 시스템 분석 방법

본 논문은 페트리 넷을 이용하여 웹 서비스 시스템의 반응 시간을 분석하고, 시스템의 병목을 찾는 방법을 소개한다. 시스템의 반응 시간은 사용자의 시스템 요구 사항 중 가장 중요한 요소 중 하나이다. 시스템의 병목은 이 부분을 보완함으로써 결국 최소의 비용으로 요구 사항을 만족하는 시스템을 설계할 수 있기 때문에, 타당성 분석에서 반드시 찾아보아야 할 요소 중 하나이다.

타당성 분석을 위한 일반적인 방법론, 즉 분석 팀을 어떻게 구성하여, 어떤 항목에 대한 조사를 하고, 어떤 정보를 추출해야 한다는 일반적인 방법론이 [10]에 소개된다. 이 연구는 타당성 분석의 일반적인 방법론을 제시하지만 본 논문이 제시하는 바와 같은 수학적 방법이 아니라는 점이 크게 다르다. 본 논문은 수학적 방법을 사용하여 분석 결과가 하나의 식으로 일목요연하게 나타난다. 또한 결과식의 구성 요소들이 시스템을 구성하는 요소들의 성능을 나타내는 변수의 형태임으로 변수에 다른 값을 대입하여 시스템을 최적화하는 변수 값을 쉽게 구할 수 있는 장점이 있다.

특정 분야에 대한 타당성 분석 방법으로 Cho[11]

는 실시간 이산 사건 시스템의 명세가 주어질 때 스케줄을 작성하는 정책과 그 정책의 타당성을 분석하는 방법을 소개하였으며, Browers[12]는 자유 비행하는 여러 대의 비행기들이 충돌하지 않도록 비행 스케줄의 타당성을 검사하는 방법을 소개한다. 또한, Manabe 등[13]은 주기적인 실시간 시스템의 타당성 검사에 대한 연구를 소개한다. 이러한 방법들은 주어진 하나의 기준을 만족하는지 판단하는데 연구의 초점을 두고 있다. 예를 들면, 비행 스케줄이 충돌을 일으키지는 않는지, 혹은 실시간 시스템에서 반응 시간이 제한 시간을 넘기지 않는지 판단하는 것이 주목적이다. 이에 반하여, 본 논문이 제공하는 방법은 시스템이 주어진 기준을 만족하는지를 판단할 뿐 아니라, 주어진 조건을 만족할 경우에는 시스템 설비 비용을 절감하는 방안을 제시하고, 만족하지 못할 경우에는 시스템의 병목을 분명히 밝힘으로써 최소의 비용으로 시스템의 성능을 향상시킬 수 있는 방안까지 제시하는 장점이 있다.

본 논문과 직접적인 관련이 깊은, 웹 서비스 시스템을 페트리 넷으로 표현하고 분석하는 방법을 다루는 논문에는 다음과 같은 것들이 있다. Thomas[4]는 WSDL로 표현된 웹 서비스 시스템을 페트리 넷으로 변환함으로써, 웹 서비스를 페트리 넷으로 표현하는 방법이 소개된다. 그러나, 페트리 넷 모형의 시간 분석은 물론, 페트리 넷을 분석하여 웹 서비스 시스템의 성질을 분석하는 방법이 결여된 단점이 있다. Hamadi[5]는 웹 서비스를 페트리 넷으로 표현하고, 페트리 넷에 연산을 적용하여 점점 더 큰 웹 서비스를 작성하는 방법이 소개된다. 이를 위하여 페트리 넷을 피연산자로 하는 연산자들이 본 논문에 정의되었다. 본 논문이 다루는 페트리 넷에도 시간을 표현하는 기능이 결여되어 있다. 작업흐름 모델에 대한 연구는 웹 서비스 조합에서 가장 중요한 부분이다. Tang[6]은 웹 서비스의 작업흐름을 페트리 넷으로 표현하고, 축소 (reduction) 규칙을 적용하여 페트리 넷을 간단히 하고, 트랜지션(transition)들의 격발 시간을 추론하는 방법을 소개한다. 제안된 시간 분석 방법은 기본적으로 작업흐름의 모든 경우를 다 따져보는 방법으로 비효율적이고 병목을 찾아주는 기능이 없다. 나아가서 처리과정에 소요되는 시간이 확률적이지 아니므로 모델에 표현된 시간이 비현실적이다. Yin[7]은 모바일 웹 서비스가 기본적으로 인증, 과금,

서비스의 순으로 이루어진다는 사실에 근거하여 웹 서비스를 표현하는 페트리 넷 모형을 구축하고, 페트리 넷 상에서 도달 테스트(reachability test)를 통하여 페트리 넷에 교착상태가 없음을 증명할 수 있다는 것을 보인다. 본 논문에 소개된 도달 테스트는 일종의 모든 경우를 다 따져보는 방법으로 비효율적인 방법이다. 더구나 시간 분석은 전혀 다루어지지 않았다. Zhang[8]은 WSDL로 표현된 웹 서비스 구조를 칼라드 넷(colored net)이라는 일종의 페트리 넷으로 변환하는 방법을 소개한다. 칼라드 넷을 시뮬레이션하는 도구가 [14]에서 제공되고 있기 때문에, 이것을 이용하여 본 논문에서 구축한 페트리 넷의 동적 성질을 분석하는 것이 가능하다. 우리의 논문은 웹 서비스 구조의 페트리 넷 모형을 구축하는 방법을 논하는 게 목적이 아니라, 이와 같은 방법으로 구축된 페트리 넷의 최소순회시간 분석 방법을 다룬다. 특히, 기존의 최소 순회 시간 분석이 상수로 표현된 시간을 다루는 데 반하여 본 논문은 시간이 혼합 확률 분포로 나타난다는 특징이 있다. Chatain[9]은 웹 서비스의 처리과정을 트랜지션(transition)으로, 선행관계를 유향간선으로 표현하여 웹 서비스를 고급 (high level) 페트리 넷으로 표현하고, 고급 페트리 넷을 전통적인 페트리 넷으로 변환하여 시스템이 수행한 처리과정의 자취를 추적하는 등 온라인 감독에 사용하는 방법이 소개된다.

페트리 넷의 트랜지션(transition)에, 트랜지션과 연관된 사건이 실행되는데 걸리는 시간을 첨부하여 놓은 것을 '시간 넷'이라 하고, '시간 넷'으로 생산 공정의 모델을 구축하고, 주기적으로 수행되는 일련의 작업이 실행되는데 걸리는 시간을 '시간 넷(timed net)'을 분석하여 알아내는 최소 순회 시간 방법이 [15]에 소개되었다. 또한, 최소 순회시간을 찾는 기존 알고리즘의 효율성을 개선하는 연구 결과가 [16]에 발표되었으며, 모든 플레이스(place)의 입력 간선과 출력 간선이 정확히 하나인 특수한 페트리 넷의 최소 순회시간을 찾는 방법이 [17]에 소개되었다. [18]에는 순회시간이 주어졌을 때, 실제 순회시간이 주어진 순회시간이 되도록 초기 마킹을 결정하는 문제에 대한 해가 소개되었다. 일반적으로 실행 시간은 확률분포를 이룸에도 불구하고, 이상의 방법들은 모두 실행 시간을 상수로 표현하는 단점이 있다. 임재걸 등[19]은 처리 과정에 소요되는 시간이 각각 정규분포,

Gamma 분포 및 χ^2 분포를 따르는 경우, 이들 분포의 합에 대한 최소 순회 시간을 분석한 바 있으나, 동일한 분포끼리만 다루는 것이 가능하다는 한계점이 있다. 본 논문은 다른 분포가 혼합하여 존재하는 경우도 다루는 혼합분포 확률 시간 넷을 이용하여 웹 서비스 시스템을 분석하는 방법을 소개한다.

2.2 혼합분포의 확률밀도함수

2006년 임 등[19] 트랜지션이 나타내는 처리과정이 완수되는데 소요되는 시간이 각각 정규분포, Gamma 분포 및 χ^2 분포를 따르는 경우에 대한 최소 순회 시간 분석을 실시한 바 있다. 그러나, 임재걸 등[19]의 연구 결과는 두 개의 트랜지션이 동일한 분포를 따르는 경우에 한정되어 있다.

본 절에서는 트랜지션이 나타내는 처리과정이 완수되는데 소요되는 시간의 분포가 각각 매개변수 λ 인 Poisson 분포와 매개변수 α, β 인 Gamma 분포인 서로 다른 분포를 따르는 경우, 이 두 분포의 혼합분포를 따르는 상황에 대한 최소 순회 시간을 분석하는 방법을 소개한다.

L. Johnson [20] 등은 특별한 개체에 대해 매개변수 λ 인 Poisson 분포에 따라 사건이 발생하는 경우와 Λ 자체의 다양성을 확률변수로 간주하여 매개변수 α 와 β 인 Gamma 분포를 따르는 경우, 두 분포에 대한 혼합분포를 제안하였다.

변수 n 이 사건 발생 횟수이고, 매개변수 λ 인 Poisson 분포를 따른다면 확률밀도함수는 다음과 같다.

$$f(n) = \frac{e^{-\lambda} \lambda^n}{n!}, \quad n=0, 1, 2, \dots \quad (1)$$

확률변수 Λ 의 확률밀도함수가

$$h(\lambda) = \frac{1}{\Gamma(\alpha)\beta^\alpha} \lambda^{\alpha-1} \exp\left(-\frac{\lambda}{\beta}\right), \quad \lambda > 0 \quad (2)$$

으로 주어진다면, Λ 는 매개변수 $\alpha > 0$ 와 $\beta > 0$ 인 Gamma 분포를 갖는다. 이때, L. Johnson 등(1985)은 식 (1)과 (2)의 혼합분포는 확률변수 Λ 와 무관한 분포로서 다음과 같이 정의하였다.

$$g(n) = \int_0^\infty \frac{e^{-\lambda} \lambda^n}{n!} \cdot \frac{1}{\Gamma(\alpha)\beta^\alpha} \lambda^{\alpha-1} \exp\left(-\frac{\lambda}{\beta}\right) d\lambda$$

$$= \binom{\alpha+n-1}{\alpha-1} \left(\frac{\beta}{\beta+1}\right)^n \left(\frac{1}{\beta+1}\right)^\alpha \quad (3)$$

식 (3)에서 $p = \frac{1}{\beta+1}$ 이면 $1-p = \frac{\beta}{\beta+1}$ 이 된다. 혼

합분포의 확률변수를 $x = a + n$ 이라 하면, $0 < p < 1$ 에서 (식 3)을 다음과 같이 쓸 수 있다.

$$f(x) = \binom{x-1}{a-1} p^a (1-p)^{x-a}, \quad x = a, a+1, \dots \quad (4)$$

식 (4)에서 확률변수 X 는 매개변수 $a=1, 2, \dots$ 와 $0 < p < 1$ 을 갖는 음이항분포(negative binomial distribution)를 갖는다.

음이항 확률변수 X 의 평균과 분산은 각각

$$\mu = \frac{a}{p} \quad \text{와} \quad \sigma^2 = \frac{a(1-p)}{p^2} \quad \text{이다.}$$

예를 들어, $p=0.95$ 이고, $a=10$ 인 음이항분포의 평균은 $\mu=10.5263$ 이고, 분산은 $\sigma^2=0.5540$ 이 된다.

3. 혼합 확률분포 시간 넷의 최소순회시간

실세계의 사건이 실행되는 데 걸리는 시간은 일반적으로 상수가 아니라 평균 얼마와 같은 확률적 수임에도 불구하고 기존의 최소 순회 시간 분석 방법은 상수만을 다룬다. 본 절에서는 시간넷의 트랜지션에 확률적 시간을 연합한 페트리 넷을 ‘확률적 시간넷’이라 하여 표 1과 같이 정의하고, 확률적 시간 넷의 최소 순회 시간을 시간을 분석하는 방법을 제안하고자 한다.

확률적 시간 넷의 예로 그림 1을 들 수 있다. 트랜지션에 연합된 지연시간은 식 (1), (2) 등과 같은 확률 분포 시간이다. 확률적 시간 넷의 최소 순회 시간 분석 방법은 기존의 방법[21]과 동일하나, 지연시간의 합을 구하는 단계에서 기존의 방법은 상수를 다루는데 반하여 확률적 시간 넷의 경우에는 통계적 확률 분포를 다룬다는 점이 다르다. 예를 들어, 그림 1의 최소 순회 시간은 다음과 같이 구한다.

초기에 각 플레이스에 놓인 토큰의 수를 나열한 벡터를 초기마킹이라 한다. 그림 1의 경우에 초기마킹 M_0 은 다음과 같다:

$$M_0 = (k, 0, k, 0, r).$$

그래프 이론에서, 그래프를 나타내는 일반적인 방법으로 인접행렬(Incidence Matrix)을 사용한다. 그림 1의 인접행렬은 다음과 같다:

$$A = \begin{matrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix} & \begin{bmatrix} -1 & 1 & -1 & 0 & -1 \\ -1 & 0 & -k & 1 & 0 \\ 1 & -1 & 1 & 0 & 1 \\ 1 & 0 & k & -1 & 0 \end{bmatrix} \end{matrix}$$

표 1. 확률적 시간 넷의 정의

확률적 시간 넷 N 은 6가지 요소로 구성된다. $N = (P, T, F, W, DT, M_0)$, 단,
 $P = \{p_1, p_2, \dots, p_m\}$ 은 플레이스(place)라는 유한 집합,
 $T = \{t_1, t_2, \dots, t_n\}$ 은 트랜지션(transition)이라는 유한 집합,
 F : 유향간선이라는 $(P \times T)$ 와 $(T \times P)$ 의 합집합의 부분 집합,
 W : $F \rightarrow \{1, 2, 3, \dots\}$ 은 유향간선의 가중치를 결정하는 무게 함수,
 DT : $T \rightarrow \{\text{Poisson}(\lambda), \text{Gamma}(\alpha, \beta)\}$, 단 λ, α, β 는 실수이며, 트랜지션의 지연 시간을 결정하는 함수,
 M_0 : $P \rightarrow \{0, 1, 2, 3, \dots\}$ 은 초기에 각 플레이스에 놓인 토큰의 수를 결정하며, 초기 마킹(marking)이라고 함.
 단, P 와 T 의 교집합은 공집합이고, P 와 T 의 합집합은 공집합이 아님.

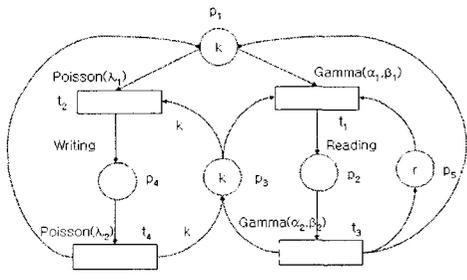


그림 1. 확률적 시간 넷의 예

[정의 1] $Ay = 0$ 을 만족하는 벡터 y 는 식 (5)의 성질을 만족하므로 S-불변(S-invariant)이라고 한다. (식 5)의 M_0 은 초기마킹, M_i 는 M_0 으로부터 도달 가능한 임의의 마킹이다.

$$y^T M_i = y^T M_0 \quad (5)$$

[그림 1]의 경우에는 다음과 같은 S-불변이 있다.

$$(0 \ 1 \ 0 \ 0 \ 1)^T = y_1$$

$$(1 \ 1 \ 0 \ 1 \ 0)^T = y_2$$

$$(0 \ 1 \ 1 \ k \ 0)^T = y_3$$

[정의 2] $A^T x = 0$ 을 만족하는 벡터 x 를 T-불변(T-invariant)이라 한다.

인접행렬 A 의 각 항 a_{ij} 는 트랜지션 i 의 격발 결과 플레이스 j 로 입력되는 토큰의 수와, 플레이스 j 에서 소모되는 토큰의 수를 나타내므로, 각 트랜지션을 T-불변의 해당 항만큼 격발하면 본래의 마킹으로 돌아오는 성질이 있다.

[정리 1] 각 트랜지션을 T-불변의 해당 항만큼 격발하면 본래의 마킹으로 돌아오는 성질이 있다.

<그림 1>의 경우에는 다음과 같은 T-불변들이 있다.

$$x_1 = (1 \ 0 \ 1 \ 0)^T,$$

$$x_2 = (0 \ 1 \ 0 \ 1)^T,$$

$$x_3 = (1 \ 1 \ 1 \ 1)^T.$$

T-불변의 정의에 의하여 정리 1의 역도 참이다. 즉, 주어진 마킹으로부터 격발을 시작하여 이 마킹으로 다시 되돌아오는 격발순서에 나타나는 트랜지션들의 회수로 구성된 벡터 x_j 는 T-불변이다.

[정리 2] 주어진 마킹으로부터 격발을 시작하여 이 마킹으로 다시 되돌아오는 격발순서에 나타나는 트랜지션들의 회수로 구성된 벡터 x_j 는 T-불변이다.

예를 들어, [그림 1]에서 t_1 과 t_3 을 차례로 격발하면 다시 [그림 1]로 되돌아온다. 따라서 이 격발순서의 격발 회수 벡터, $(1 \ 0 \ 1 \ 0)$ 은 T-불변이다.

인접행렬의 음수 항만 절대 값으로 바꾸고, 나머지 항은 0으로 대치하여 얻은 행렬, 즉 각 플레이스로 들어오는 간선의 무게만 표기한 행렬을 A 행렬이라 하며, [그림 1]의 경우에는 다음과 같다:

$$A^- = \begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & k & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

[그림 1]의 각 트랜지션에 연관된 지연 시간을 나타내는 대각행렬 D 는 다음과 같다.

$$D = \begin{matrix} & t_1 & t_2 & t_3 & t_4 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix} & \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \end{matrix}$$

최소 순회시간을 찾는 식은 Murata[21]에 소개되어 있고, (식 6)과 같다. 단, x 는 x_3 과 같이 양수 항으로 구성된 T-불변이다.

최소 순회 시간

$$= \text{Max} \{y_k^T (A^-)^T D x / y_k^T M_0\} \quad (6)$$

[그림 1]의 시간 넷에 식 (6)을 적용하면, 다음과 같다.

$$\begin{aligned}
 (A^-)^T Dx &= [d_1 + d_2 \ d_3 \ d_1 + kd_2 \ d_4]^T \\
 y_1 &= (01001)^T \text{ 일때,} \\
 y_1^T (A^-)^T Dx &= d_1 + d_3, \\
 y_1^T M_0 &= (01001)(k \ 0 \ k \ 0 \ r)^T = r \text{ 이므로,} \\
 y_1^T (A^-)^T Dx / y_1^T M_0 &= (d_1 + d_3)/r \text{ 이다.} \\
 y_2 &= (11010)^T \text{ 일때,} \\
 y_2^T (A^-)^T Dx &= d_1 + d_2 + d_3 + d_4, \\
 y_2^T M_0 &= k \text{ 이므로,} \\
 y_2^T (A^-)^T Dx / y_2^T M_0 &= (d_1 + d_2 + d_3 + d_4)/k \\
 &\text{이다.} \\
 y_3 &= (011k0)^T \text{ 일때,} \\
 y_3^T (A^-)^T Dx &= d_1 + d_3 + k(d_2 + d_4), \\
 y_3^T M_0 &= k \text{ 이므로,} \\
 y_3^T (A^-)^T Dx / y_3^T M_0 &= d_2 + d_4 + (d_1 + d_3)/k \text{이다.}
 \end{aligned}$$

따라서 최소 순회 시간은 다음과 같다.

$$\begin{aligned}
 \text{Max} \{ (d_1 + d_3)/r, (d_1 + d_2 + d_3 + d_4)/k, \\
 d_2 + d_4 + (d_1 + d_3)/k \} = d_2 + d_4 + (d_1 + d_3)/k
 \end{aligned}$$

즉, 모든 트랜지션을 정확히 한 번씩 격발하고 초기 마킹으로 돌아오는데 걸리는 시간은 $d_2 + d_4 + (d_1 + d_3)/k$ 이다. 기존의 연구에서는 d_1, \dots, d_4 가 각각 상수임으로 계산 결과도 상수가 된다. 그러나 혼합 확률 분포 시간 넷의 경우에는 (식 3)을 이용하여 다음과 같이 설명할 수 있다:

$$\begin{aligned}
 & \text{Poisson}(\lambda_1) + \text{Poisson}(\lambda_2) \\
 & + \{ \text{Gamma}(a_1, \beta_1) + \text{Gamma}(a_2, \beta_2) \} / k \quad (7)
 \end{aligned}$$

(식 7)에서 Poisson 분포의 합은 Poisson 분포가 되고, Gamma 분포의 합은 Gamma 분포가 된다. 따라서, 혼합분포는 식 (4)에서와 같이 음이항분포가 되며, 이 분포를 이용하여 얻는 확률값은 α 및 β 값에 따라 다르다.

자연 시간이 Poisson 분포와 Gamma 분포의 혼합 분포를 이룰 경우를 고려하여 보자. 두개의 독립적인 트랜지션 t_2 와 t_4 및 t_1 과 t_3 에 의해 자연 시간이 각각 Poisson과 Gamma분포를 따른다면 그림 1의 분석 결과는 식 (4)에 의해 결정되는데, 혼합분포 $f(x)$ 의 확률값은 사용자가 요구하는 처리시간의 95% 확률 범위에서 10.5263 단위시간이라면 이 경우에는 사용자의 요구를 만족하지 못한다는 결론을 내린다.

실험자료의 수가 많아지면 Poisson 분포와 Gamma 분포는 정규분포에 수렴한다는 확률분포로서의 성질

을 가지고 있으며, 이들의 혼합분포인 음이항분포도 마찬가지이다.

4. 웹 서비스 시스템의 혼합 확률 분포 시간 넷 모형

웹 서비스의 페트리 넷 모형은 [7]에 소개된 바와 같이 처리과정을 트랜지션으로, 웹 서비스의 상태나 자원을 플레이스로, 그리고 처리 과정들의 선후 관계를 간선으로 나타냄으로써 구축한다.

[22]에 의하면 웹 서비스 컨텍스트(context)의 생명주기는 다음과 같다. 컨텍스트란 단말기의 유형이나 사용자의 위치 정보와 같은 사용자의 환경을 지칭하는 용어이다. 컨텍스트는 고객의 사이트에서 태어난다. 고객이 관련된 컨텍스트 정보를 수집하여 SOAP 요구 메시지에 컨텍스트 헤더블럭으로 삽입함으로써 컨텍스트의 생명주기가 시작된다. 웹 서비스 플랫폼에 의하여 요구 메시지가 접수되면, 컨텍스트 프레임워크에 의하여 컨텍스트가 추출되고 호출된 웹 서비스에게 현재 컨텍스트로 제공된다. 웹 서비스는 수행되는 동안, 현재 컨텍스트를 프레임워크가 제공하는 컨텍스트 API를 이용하여 접근하고 변경한다.

웹 서비스가 수행되는 동안 다른 웹 서비스를 호출하면 현재 컨텍스트가 자동적으로 나가는 요구 메시지에 삽입된다. 이에 대한 응답 메시지도 컨텍스트 정보를 포함할 수 있다. 그러한 경우에는 필요한 컨텍스트 데이터 부분을 추출하여 현재 컨텍스트에 삽입한다. 웹 서비스가 종료할 때에는, 응답 메시지에 현재 컨텍스트를 삽입하여 호출자에게 되돌려준다. 호출자가 고객이라면 되돌아온 컨텍스트의 일부를 자신의 컨텍스트 정보에 삽입할 수도 있다. 나아가서, 고객은 되돌아온 컨텍스트를 참조하여 웹 서비스의 응답을 조절할 수도 있다.

이와 같은 컨텍스트의 생명주기는 웹 서비스의 처리 과정과 매우 유사하다. 따라서 컨텍스트의 생명주기를 나타내는 모형으로부터 시스템의 반응 시간을 분석할 수 있다. [7]에 소개된 방법에 의하여 처리과정을 트랜지션으로, 상태나 자원을 플레이스로 나타내면, 컨텍스트의 생명주기에 대한 페트리 넷 모델은 그림 2와 같다. 그림 2는 CPNtools[23]라는 페트리 넷 모형 구축 및 시뮬레이션 소프트웨어의 양식에 따라 표현한 것이다. Client라고 이름이 붙은 플레이스(페트리 넷 그림에서 타원(혹은 원)을 place라 함)

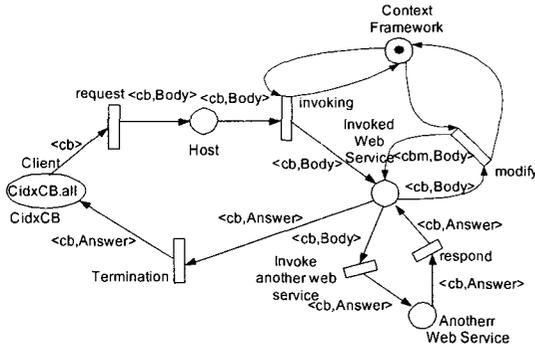


그림 2. 웹 서비스 과정의 페트리 넷 모형

표 2. 그림 2에 사용되는 변수 선언문

```

Cid: {1..100} ; // a client's identity number
Location = product real*real; // coordinate
Action, Role, Body, Answer : string;
Time: integer;
ElementOfContext: {null, Location, Action, Time, Role};
ELE: ElementOfContext
CB = product of ELE*ELE*ELE*ELE*ELE;
CidxCB = product of Cid * CB ;
Var cb, cbi, cbm : CidxCB ; // cb stands for
// context block, cbi is another cb,
// cbm is a modified cb.
    
```

안에 보이는 CidxCB.all은 CPNtools가 제공하는 함수로 CidxCB의 모든 원소를 나열하는 기능이 있다. 그림 2의 하단부를 보면 CidxCB는 Product of Cid * CB로 정의되어 있는데, 이는 CidxCB라는 이름의 데이터 유형이 Cid와 CB의 카티션 곱임을 나타낸다. Cid는 1부터 100 사이의 정수로 구성된 데이터 형으로, Cid는 Client ID, 즉 고객의 식별번호를 의미한다. CB는 ELE의 5승으로 정의되어 있으며, ELE는 null, Location, Action, Time, Role로 구성된 데이터 형으로 정의되어 있다. 즉, CB는 null, Location, ... 등을 요소로 하는 5-tuple이다. Location, Action 등은 컨텍스트를 구성하는 요소들이고, CB는 Context Block을 의미한다.

플레이스 Client의 출력 간선의 레이블인 <cb>는 그림 2의 하단부에 데이터 형이 CidxCB인 변수로 선언되어 있다. 즉 cb는 고객식별번호와 context block의 순서쌍을 값으로 갖는 변수이다. 페트리 넷 분야에서는 변수 값이 될 수 있는 토큰을 colored 토큰이라 하고, 플레이스 'Context Framework'에 있는

까만 점을 그냥 토큰이라고 한다. 'Context Framework'에도 다양한 API들이 있어서 각 API들을 구분하여 나타내고 싶을 때에는 colored 토큰을 사용하지만, API 전체를 하나의 추상적인 객체로 고려할 때에는 그림 2처럼 color가 없는 토큰으로 간단히 표현할 수 있다.

어떤 트랜지션은 모든 입력 플레이스에 충분한 수의 토큰이 놓여 있을 때 fire할 수 있다. 어떤 플레이스 a와 트랜지션 b가 유향간선 (a, b)로 연결되어 있을 때, 플레이스 a에 (a, b)의 레이블보다 적지 않은 수의 토큰이 있으면 플레이스 a는 트랜지션 b를 fire하기에 충분한 수의 토큰을 가지고 있다고 한다. 어떤 트랜지션 b의 fire는 모든 입력 플레이스 a에서 간선 (a, b)의 레이블 만큼의 토큰을 삭제하고, 모든 출력 플레이스 c에 간선 (b, c)의 레이블만큼의 토큰을 추가하여 놓는다.

그림에서는 트랜지션 'request'가 fire할 수 있으며, fire 결과 플레이스 'Host'에 <cb, Body>라는 토큰을 생성하여 놓는다. cb는 앞에서 설명한 바와 같이 고객 식별번호와 컨텍스트 블록으로 구성되며, Body는 문자열로 선언되어 있다. 트랜지션 'request'는 고객이 메뉴를 선택함에 따라 SOAP 메시지를 구성하는 과정을 흉내낸다.

고객이 사용하는 호스트가 웹 서비스를 호출하는 과정을 흉내내는 트랜지션이 'invoking'이다. 이때 'Context Framework'의 API를 이용하여 'current context'를 추출한다. 컨텍스트 블록은 'Context Framework'의 도움을 받아 변경될 수 있다. 트랜지션 'modify'가 이것을 흉내 내며, 0번 이상 여러 번 실행될 수 있다. 웹 서비스는 다른 웹 서비스에게 서비스를 요청할 수 있는데 이것을 흉내내는 트랜지션이 'Invoke another web service'이며, 0번 이상 여러 번 실행된 후 결국 고객을 위한 답을 구성하여 트랜지션 'Termination'을 통하여 고객에게 전달한다. 본 모델에서는 Answer도 Body와 같이 String으로 선언되어 있다. 그림 2의 모델은 각 플레이스에 데이터 형을 부착시키고 변수를 몇 개 더 선언하여 사용하면 CPN tools로 시뮬레이션이 가능한 colored 페트리 넷 모형이 된다.

웹 서비스 시스템의 시간 분석을 위하여 그림 2를 시간 넷으로 변형하면 그림 3과 같다. 각 트랜지션과 플레이스에 식별자를 $t_1, t_2, \dots, t_B, p_1, p_2, \dots, p_B$ 라고 부

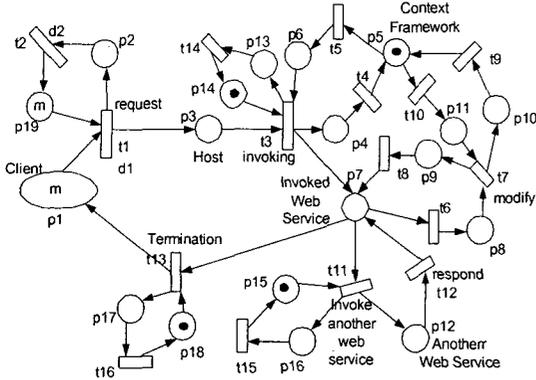


그림 3. 그림2에 대한 시간 넷

치고, 트랜지션 t_i 가 표현하는 처리과정이 수행되는데 소요되는 시간을 d_i 라고 한다. 그림 3에는 d_1 과 d_2 만 표시하고 나머지는 생략하였다. 기존의 연구에서는 d_i 가 상수였지만, 본 논문에서는 Poisson 분포나 Gamma분포일 수도 있다는 점이 다른 점이다. 트랜지션 t_{14}, t_{15}, t_{16} 은 각각 요구 메시지의 전송 시간, 다른 웹 서비스를 호출하는데 걸리는 시간, 그리고 웹 서비스의 결과를 고객에게 전달하는데 소요되는 전송시간을 나타내기 위한 트랜지션이다.

그림 3의 최소 순회 시간 분석은 다음과 같이 진행된다. 그림 3의 초기 마킹, M_0 을 다음과 같이 구한다.

$$M_0 = (m \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ m).$$

플레이스 p1의 m은 고객의 수를 나타내며, m은 고객 각자가 자기의 단말기를 사용한다는 것을 나타낸다.

그리고 그림 3에 대한 인접행렬 A를 구하고, $A^T x = 0$ 을 만족하는 벡터 x , 즉 T-불변을 구한다. T-불변 중 모든 항이 양의 정수인 것은 x_4 이다. x_4 는 h, j, k가 임의의 양의 정수일 때 $x_4 = h * x_1 + j * x_2 + k * x_3$ 로 구한다.

t_1	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}
t_1	-1	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_2	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+1
t_3	0	0	-1	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_4	0	0	0	-1	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_5	0	0	0	0	-1	+1	0	0	0	0	0	0	0	0	0	0	0	0	0
t_6	0	0	0	0	0	-1	+1	0	0	0	0	0	0	0	0	0	0	0	0
t_7	0	0	0	0	0	0	-1	+1	0	0	0	0	0	0	0	0	0	0	0
t_8	0	0	0	0	0	0	0	-1	+1	0	0	0	0	0	0	0	0	0	0
t_9	0	0	0	0	0	0	0	0	-1	+1	0	0	0	0	0	0	0	0	0
t_{10}	0	0	0	0	0	0	0	0	0	0	-1	+1	0	0	0	0	0	0	0
t_{11}	0	0	0	0	0	0	0	0	0	0	0	0	-1	+1	0	0	0	0	0
t_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	+1	0	0	0
t_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	+1	0
t_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
t_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1

x_4 로부터 알 수 있는 웹 서비스 시스템의 성질은

고객이 원하는 답을 구할 때까지 컨텍스트를 변경하는 작업이 여러 번 실행될 수도 있고, 다른 웹 서비스를 여러 번 호출할 수도 있으며, 변경 작업이나 다른 웹 서비스 호출이 전혀 없이 고객이 호출한 웹 서비스만 실행하여 답을 제공하는 작업이 여러 번 수행될 수도 있다는 것이다.

$$x_1 = (1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1)^T,$$

$$x_2 = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0)^T,$$

$$x_3 = (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T,$$

$$x_4 = (h \ h \ h \ h \ h \ k \ k \ k \ k \ k \ j \ j \ h \ h \ j \ h)^T.$$

그림 3에 대한 A행렬을 구하면 다음과 같다.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}
t_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_3	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
t_4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
t_7	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
t_8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
t_9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
t_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

그림 3에 대한 D행렬을 구하면 다음과 같다.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
t_1	d_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_2	0	d_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_3	0	0	d_3	0	0	0	0	0	0	0	0	0	0	0	0	0
t_4	0	0	0	d_4	0	0	0	0	0	0	0	0	0	0	0	0
t_5	0	0	0	0	d_5	0	0	0	0	0	0	0	0	0	0	0
t_6	0	0	0	0	0	d_6	0	0	0	0	0	0	0	0	0	0
t_7	0	0	0	0	0	0	d_7	0	0	0	0	0	0	0	0	0
t_8	0	0	0	0	0	0	0	d_8	0	0	0	0	0	0	0	0
t_9	0	0	0	0	0	0	0	0	d_9	0	0	0	0	0	0	0
t_{10}	0	0	0	0	0	0	0	0	0	d_{10}	0	0	0	0	0	0
t_{11}	0	0	0	0	0	0	0	0	0	0	d_{11}	0	0	0	0	0
t_{12}	0	0	0	0	0	0	0	0	0	0	0	d_{12}	0	0	0	0
t_{13}	0	0	0	0	0	0	0	0	0	0	0	0	d_{13}	0	0	0
t_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	d_{14}	0	0
t_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d_{15}	0
t_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d_{16}

$Ay = 0$ 을 만족하는 S-불변을 구하면 다음과 같다.

$$(0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0)^T = y_1$$

$$(0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0)^T = y_2$$

$$(0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0)^T = y_3$$

$$(0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1)^T = y_4$$

$$(0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T = y_5$$

$$(1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T = y_6$$

양의 T-불변 x_4 를 x 로 하여 $(A^-)^T D x$ 를 구하면 다음과 같다.

$$(A^-)^T D x = (hd_1\ hd_2\ hd_3\ hd_4\ hd_5 + kd_{10}\ hd_3\ kd_6 + jd_{11} + hd_{13}\ kd_7\ kd_8\ kd_9\ kd_7\ jd_{12}\ hd_{11}\ hd_3\ jd_{15}\ jd_{15}\ hd_{16}\ hd_{13})^T$$

앞에서 구한 S-불변 각각에 대하여 $y_k^T (A^-)^T D x / y_k^T M_0$ 를 다음과 같이 구한다.

$$(00000000000000000000110)^T = y_1 \text{의 경우,}$$

$$y_1^T (A^-)^T D x = h(d_{16} + d_{13}), \quad y_1^T M_0 = 1,$$

그러므로, $y_1^T (A^-)^T D x / y_1^T M_0 = h(d_{16} + d_{13})$.

$$(000000000000000000011000)^T = y_2 \text{의 경우,}$$

$$y_2^T (A^-)^T D x = j(d_{11} + d_{15}), \quad y_2^T M_0 = 1,$$

그러므로, $y_2^T (A^-)^T D x / y_2^T M_0 = j(d_{11} + d_{15})$.

$$(000000000000000001100000)^T = y_3 \text{의 경우,}$$

$$y_3^T (A^-)^T D x = h(d_{14} + d_3), \quad y_3^T M_0 = 1,$$

그러므로, $y_3^T (A^-)^T D x / y_3^T M_0 = j(d_{14} + d_3)$.

$$(0100000000000000000001)^T = y_4 \text{의 경우,}$$

$$y_4^T (A^-)^T D x = h(d_1 + d_2), \quad y_4^T M_0 = m,$$

그러므로, $y_4^T (A^-)^T D x / y_4^T M_0 = h(d_1 + d_2)/m$.

$$(00011100001100000000)^T = y_5 \text{의 경우,}$$

$$y_5^T (A^-)^T D x = h(d_3 + d_4 + d_5) + k(d_7 + d_9 + d_{10}),$$

$$y_5^T M_0 = 1, \text{ 그러므로 } y_5^T (A^-)^T D x / y_5^T M_0 =$$

$$h(d_3 + d_4 + d_5) + k(d_7 + d_9 + d_{10}).$$

$$(10100011100100000000)^T = y_6 \text{의 경우,}$$

$$y_6^T (A^-)^T D x = h(d_1 + d_3 + d_{13}) + k(d_6 + d_7 + d_8) + j(d_{11} + d_{12}).$$

$$y_6^T M_0 = m, \text{ 그러므로}$$

$$y_6^T (A^-)^T D x / y_6^T M_0 =$$

$$\{h(d_1 + d_3 + d_{13}) + k(d_6 + d_7 + d_8) + j(d_{11} + d_{12})\} / m.$$

따라서 그림 3의 최소 순회 시간은 다음과 같다.

$$\text{Max} \{y_k^T (A^-)^T D x / y_k^T M_0\} = \text{Max} [h(d_{16} + d_{13}),$$

$$j(d_{11} + d_{15}), \quad j(d_{11} + d_3), \quad h(d_1 + d_2)/m,$$

$$h(d_3 + d_4 + d_5) + k(d_7 + d_9 + d_{10}),$$

$$\{h(d_1 + d_3 + d_{13}) + k(d_6 + d_7 + d_8) + j(d_{11} + d_{12})\} / m].$$

5. 모델의 이용 방법

그림 3의 각 트랜지션에 연합한 지연 시간의 의미

는 다음과 같다. d_1 은 서비스 요구 메시지를 구성하는 시간, d_2 는 시스템의 인터페이스를 통하여 고객이 메뉴를 선택하는 데 걸리는 지연, d_3 는 요구 메시지가 원격지에 있는 웹 서비스 제공 사이트에 전달되어 웹 서비스가 호출되기까지 소요되는 시간, $d_4 + d_5$ 는 웹 서비스가 호출될 때 컨텍스트 블록을 분리하여 current context를 형성하는데 걸리는 시간, $d_6 + d_7 + d_8$ 은 웹 서비스가 current context를 변경하는데 걸리는 시간, $d_9 + d_{10}$ 은 컨텍스트 프레임워크가 current context를 변경하는 서비스를 실행하는데 걸리는 시간, $d_{11} + d_{12}$ 는 다른 웹 서비스의 서비스를 받는 데 걸리는 시간, d_{13} 은 웹 서비스 결과를 고객에게 전달하는데 소요되는 시간, d_{14} 는 고객의 단말기에서 요구 메시지가 웹 서비스 시스템으로 전송되는데 걸리는 시간, d_{15} 는 웹 서비스로부터 호출된 다른 웹 서비스가 서비스를 제공하는 데 걸리는 시간이며, d_{16} 은 고객이 호출한 웹 서비스가 결과를 전송하는데 걸리는 시간이다.

따라서 고객이 메뉴를 선택하는데 걸리는 시간 d_2 는 평균 8000 단위시간(밀리초)의 Poisson분포, 서비스 요구 메시지를 구성하는 시간 d_1 은 평균 10 단위시간의 Poisson분포, SOAP의 헤드를 조작하는 시간 d_3 은 평균 20 단위시간의 Poisson분포, d_7 은 평균 10 단위 시간의 Poisson분포, d_{14} 와 d_{16} 은 전송 시간 이므로 각각 평균 100 단위시간인 Gamma분포, 다른 웹 서비스의 서비스를 받는데 소요되는 시간인 d_{15} 는 평균 240 단위 시간의 Gamma분포, d_{11} 은 평균 10 단위시간의 Gamma분포, d_{13} 은 상수 2 단위 시간, 나머지는 모두 무시할 만큼 작은 시간, 즉 0이라고 할 수 있다. m 은 고객의 수이므로 10 명이라고 하고, $h=1, j=2, k=3$ (즉, 웹 서비스 1회에 다른 웹 서비스 호출이 2번, 컨텍스트 변경이 3회 발생하는 상황을 반영함)이라고 하자. 그러면,

$$\text{최소순회시간} = \text{Max} [h(d_{16} + d_{13}), \quad j(d_{11} + d_{15}),$$

$$j(d_{11} + d_3), \quad h(d_1 + d_2)/m, h(d_3 + d_4 + d_5) + k(d_7 + d_9 + d_{10}),$$

$$\{h(d_1 + d_3 + d_{13}) + k(d_6 + d_7 + d_8) + j(d_{11} + d_{12})\} / m]. \quad (8)$$

즉, 최소순회시간은 $(d_1 + d_2)/10$ 으로서

$$\text{Max}\{102, 500, 240, 801, 20, 30, 8.2\} = 801$$

이 된다. 최소 순회시간이 d_2 에 의하여 결정된다는 것은 고객이 메뉴를 선택하는 시간이 웹 서비스의 반응 시간보다 더 길다는 것을 의미한다. 따라서 웹

서비스의 반응 시간이 고객의 요구를 충족시키기에 충분하다는 결론을 내릴 수 있다.

만일 고객의 수를 20명이라고 하면 최소순회시간은 $2(d_{11} + d_{15})$ 로서 500이 된다.

Poisson분포를 따르는 $h(d_1 + d_2)/m$ 이나 Gamma 분포를 따르는 $j(d_{11} + d_{15})$ 은 자료의 수가 커지면 중심극한정리에 의해 모두 정규분포에 수렴하는 성질을 가지고 있는데, 이것을 근거로 이들 순회시간의 분포적 성질을 비교할 수 있다.

5.1 최소순회시간이 801인 경우

식 (8)의 최대 값은 801 임으로, 고객의 수가 10명 일 때, 고객이 메뉴를 선택하는 시간이 시스템의 응답시간보다 빠르다는 것을 알 수 있다. 시스템의 응답 속도가 어느 정도 필요 이상으로 빠르지 알아보기 위하여 식 (8)의 두 번째 큰 값인 500에 해당하는 다른 웹 서비스의 응답이 도착하는데 걸리는 시간의 분포와 801에 해당하는 고객의 메뉴 선택 시간의 분포를 비교하면, 표준오차가 각각 751.25로 동일하다면 그림 4와 같다.

그림 4의 검은색 부분은 시스템이 필요 이상으로 빠르게 응답하는 경우에 해당하며, 이 경우의 확률을 아래의 계산과정을 거쳐 표준정규분포표를 이용하여 구할 수 있다. 0.1586이라는 확률이 너무 크다고 생각이 되면 가입자를 늘려 더 많은 고객을 서비스함으로써 수입을 크게 하든지 아니면 시스템을 구성하는 장비의 사양을 낮춤으로써 시스템 구현에 드는 비용을 절감할 수 있다.

$$1 - 2 \times P(z \leq \frac{650.25 - 500}{751.25})$$

$$= 1 - 2 \times P(z \leq 0.2) = 0.1586$$

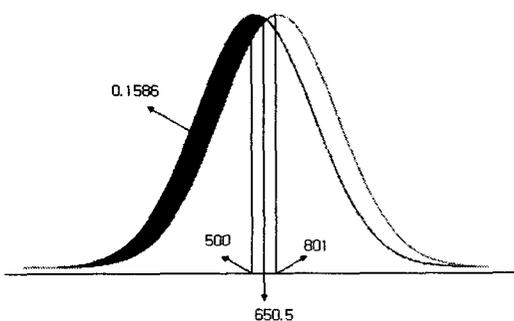


그림 4. 시스템의 서비스 시간과 고객의 메뉴선택시간의 확률 분포 (최소순회시간이 801인 경우)

5.2 최소순회시간이 500인 경우

고객의 수를 20명으로 늘릴 경우 (식 8)의 결과는 500 단위시간이 된다. 따라서 고객의 메뉴 선택 시간보다 시스템의 응답시간이 더 길다는 것을 알 수 있다. 고객이 메뉴 선택을 위하여 소비하는 시간보다 응답시간이 너무 길면 고객의 불만을 초래하게 된다. 따라서 응답시간이 얼마나 더 긴지 분석해볼 필요가 있다.

만약 여러 번의 실험에서 위와 같은 결과를 평균 값으로 얻었다고 할 때, 표준오차가 각각 751.25로 동일하다면 고객의 수 20명에 대한 최소순회시간에 대한 분포를 그림 5와 같이 그릴 수 있다. 이 경우 요구사항을 만족시키지 못할 확률은 그림의 검은색 부분으로 아래의 계산과정을 거쳐 표준정규분포표를 이용하여 구할 수 있다. 이로부터 이 시스템으로 20명의 고객을 서비스한다면 95% 이상의 경우에 고객을 만족시킬 수 있으며, 고객의 메뉴 선택시간보다 응답시간이 더 길 확률은 0.0478이라는 것을 알 수 있다.

$$1 - 2 \times P(z \leq \frac{449.9 - 400.25}{751.25})$$

$$= 1 - 2 \times P(z \leq 0.06) = 0.0478$$

6. 결 론

실세계에서는 어떤 사건이 실행되는 데 소요되는 시간이 일반적으로 상수가 아니라 통계적 확률분포를 이룬다. 그러나 기존의 최소 순회 시간 분석에서는 시간 넷의 트랜지션에 연함된 지연 시간이 상수인 경우만 다루고 있다. 따라서 기존의 방법으로는 다양

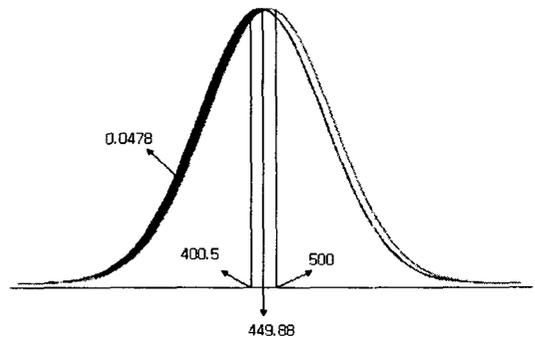


그림 5. 시스템의 서비스 시간과 고객의 메뉴선택시간의 확률 분포 (최소순회시간이 500인 경우)

한 확률모형에 의한 확률적인 분석 기준을 뒷받침할 수가 없다.

본 논문은 시간 넷의 트랜지션에 연합된 자연 시간이 두 개의 서로 다른 확률분포를 하는 경우, 즉, 혼합분포를 하는 확률적 시간 넷을 제안하고, 이것을 이용한 최소 순회 시간 분석 방법을 소개하였다. 제안하는 방법은 기존의 최소 순회 시간 분석 방법과 비슷하지만 제안하는 방법의 분석 결과는 기존의 방법의 결과와 크게 다르다.

확률적인 분석 결과는 일상적으로 사용되는 퍼지 문장으로 표현된 사용자의 요구 사항을 분석하는 데에도 적용될 수 있다. 예를 들어, 시스템 요구 사항 중에 “시스템의 응답시간이 고객의 메뉴선택 시간 8,000 단위시간보다 반드시 적어야 한다.”라는 사항이 들어 있다면 5절에서 다룬 두 가지 경우 중 고객의 수를 10명으로 하는 경우를 선택해야 한다. 그러나 시스템 요구 사항이 “일반적으로 시스템의 응답시간이 고객의 메뉴선택 시간 8,000 단위시간보다 적어야 한다.”라고 기술되었다면 5절에서 보인 두 가지 경우의 분석 결과에 의하여 고객의 수를 20명으로 잡아도 된다는 것을 알 수 있다.

제안하는 방법이 아닌 기존의 방법으로 5절의 경우를 분석하면 고객의 메뉴 선택 시간인 400.5 단위 시간이 시스템의 응답시간인 500 단위 시간보다 분명히 더 작기 때문에 시스템 요구사항이 비록 “일반적으로 시스템의 응답시간이 고객의 메뉴선택 시간 8,000 단위시간보다 적어야 한다.”라고 기술되어 있다고 하더라도 시스템의 요구사항을 만족시키지 못한다는 결론을 내릴 것임으로, 이 예를 통하여 제안하는 방법이 훨씬 더 실용적인 것임을 알 수 있다.

본 논문은 기존의 최소 순회 시간 분석을 자연 시간이 확률분포를 나타내는 경우로 확장하여 실용성을 보장하고, 본 논문이 제안한 혼합 분포 시간 넷을 웹 서비스 시스템 분석에 적용하는 예를 보였다. 향후에는 확률적인 자연 시간을 바탕으로 퍼지 시스템을 분석하는 수학적 방안을 고안하고자 한다.

참 고 문 헌

[1] <http://www.w3.org/TR/soap/>
 [2] <http://www.w3.org/TR/wsdl>
 [3] <http://www.uddi.org/>
 [4] J.P. Thomas, M. Thomas, and G. Ghinea,

“Modeling of Web services flow,” *Proceedings of the IEEE International Conference on E-Commerce*, pp.391-398, 24-27 June 2003.
 [5] R. Hamadi and B. Benatallah, “A Petri Net-based Model for Web Service Composition,” *Proceedings of Fourteenth Australasian Database Conference (ADC2003)*, Adelaide, Australia. Feb. 2003. *Conferences in Research and Practice in Information Technology*, Vol. 17. Schewe, K.-D. and Zhou, X., Eds., pp. 191-200. available on http://crpit.com/confpapers/CRPI_TV17Hamadi.pdf
 [6] Y. Tang, L. Chen, K. He, and N. Jing, “SRN: an extended Petri-net-based workflow model for Web service composition,” *Proceedings of IEEE International Conference on Web Services*, pp. 591-599, 6-9 July 2004.
 [7] H. Yin, Z. Tan, C. Lin, Y. Jiang, and F. Qiu, “Modeling mobile Web services flow using generalized stochastic high-level Petri net,” *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, pp. 4953-4958, 10-13 Oct. 2004.
 [8] J. Zhang, C. Chang, J. Chung, and S. Kim, “WS- Net: a Petri-net based specification model for Web services,” *Proceedings. IEEE International Conference on Web Services*, pp. 420-427, 6-9 July 2004.
 [9] T. Chatain and C. Jard, “Models for the supervision of Web services orchestration with dynamic changes,” *Proceedings of Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop. AICT/SAPIR/ELETE 2005*, pp. 446-451, 17-20 July 2005.
 [10] C.M. Khoong and Y.W. Ku,, “A holistic feasibility study framework for decision systems,” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, Issue. 1, pp. 100-106, Jan. 1994.
 [11] S.M. Cho and T.G. Kim, “Analysis of feasi-

bility for real time simulation of RT-DEVs models," *IEEE International Conference on Systems, Man, and Cybernetics* 2001. Tucson, AZ, USA, Vol. 5, pp. 3069-3074, Oct. 7-10 2001.

[12] K.L. Bowers, "Determining the feasibility of a flight profile in a Free Flight environment," *Proceedings of Digital Avionics Systems Conference*, 1996 (15th AIAA/IEEE), Atlanta, GA USA, pp. 81-86, Oct. 27-31 1996.

[13] Y. Manabe and S. Aoyagi, "A feasibility decision algorithm for rate monotonic scheduling of periodic real-time tasks," *Proceedings of Real-Time Technology and Applications Symposium*, 1995, Chicago, IL USA, pp. 212-218, May 15-17 1995.

[14] <http://www.daimi.au.dk/designCPN/>

[15] Hillion and Hervé P., "Timed Petri Nets and Application to Multi-Stage Production Systems," *Lecture Notes in Computer Science*, Vol. 424; *Advances in Petri Nets* 1989, Berlin, Germany, Springer-Verlag, pp. 281-305, 1990.

[16] A. Dasdan and R.K. Gupta, "Faster maximum and minimum mean cycle algorithms for system performance analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, No. 10, pp. 889-899, 1998.

[17] M. Nakamura and M. Silva, "Cycle time computation in deterministically timed weighted marked graphs," *Proceedings. ETFA '99. 7th IEEE International Conference on Emerging Technologies and Factory Automation 1999*, Vol. 2. pp. 1037-1046, Oct. 18-21, 1999.

[18] J. Rodriguez-Beltran and A. Ramfrez-Trevino, "Minimum initial marking in timed marked graphs," *2000 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, pp. 3004-3008, Oct. 8-11, 2000.

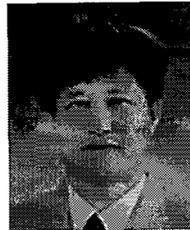
[19] 임재걸, 심규박, "확률적 시간 넷과 최소 순회 시간분석," *멀티미디어학회 논문집*, 2006년 5월호, pp.671-680, 2006.

[20] L. Johnson, S. Kotz, *Encyclopedia of Statistical Sciences*, Vol. 5, John Wiley & Sons, 1985, ISBN 0-471-05552-2.

[21] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, Vol. 77. No. 4, pp. 541-580, April 1989.

[22] M. Keidl and A. Kemper, "Towards Context-Aware Adaptable Web Services," *The Proceeding of the Thirteenth International World Wide Web Conference*, pp. 55-65, available on <http://www2004.org/proceedings/docs/2p55.pdf>.

[23] <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>



임 재 걸

1981년 동국대학교 전자계산학과 졸업
 1987년 일리노이대학교 시카고 캠퍼스 컴퓨터과학 석사
 1990년 일리노이대학교 시카고 캠퍼스 컴퓨터과학 박사
 1992년~현재 동국대학교 과학기

술대학 컴퓨터멀티미디어 학과 교수
 관심분야 : 시스템 설계 및 분석, 인공지능, 페트리 넷 이론 및 응용.



심 규 박

1984년 동국대학교 통계학과 졸업
 1986년 동국대학교 대학원 통계학과 이학석사
 1993년 동국대학교 대학원 통계학과 이학박사
 1994년~현재 동국대학교 과학기

술대학 정보통계학과 교수
 관심분야 : 전산통계, 신뢰도검정, 통계 자료분석.



도 재 수

1991년 경북대학교 전자공학과 졸업
 1994년 홋카이도대학 대학원 전자공학과 공학석사
 1997년 홋카이도대학 대학원 전자공학과 공학박사
 1997년~현재 동국대학교 과학기

술대학 컴퓨터멀티미디어학과 부교수
 관심분야 : 영상신호처리, 영상압축, 웨이브릿변환.